Methods in Ecology and Evolution

Deep Learning and Satellite Imagery predict genetic diversity and differentiation

Journal:	*Methods in Ecology and Evolution*
Manuscript ID	MEE-21-09-754
Manuscript Type:	Research Article
Date Submitted by the Author:	30-Sep-2021
Complete List of Authors:	Kittlein, Marcelo; Universidad Nacional de Mar del Plata, Biology Mora, Matías; Departamento de Biología. Instituto de Investigaciones Marinas y Costeras (IIMyC). Facultad de Ciencias Exáctas y Naturales Universidad Nacional de Mar del Plata. Consejo Nacional de Investigaciones Científica y Técnicas (CONICET), Mar del Plata, Argentina. Mapelli, Fernando; Grupo de Genética y Ecología en Conservación y Biodiversidad (GECOBI), División Mastozoología. Museo Argentino de Ciencias Naturales ``Bernardino Rivadavia'', (CONICET), Ciudad de Buenos Aires, Argentina. Austrich, Ailín; Departamento de Biología. Instituto de Investigaciones Marinas y Costeras (IIMyC). Facultad de Ciencias Exáctas y Naturales Universidad Nacional de Mar del Plata. Consejo Nacional de Investigaciones Científica y Técnicas (CONICET), Mar del Plata, Argentina. Gaggiotti, Oscar; University of St Andrews, Scottish Oceans Institute;
Keywords:	biodiversity prediction, convolutional neural networks, coastal dunes, Ctenomys australis, Deep Learning, genetic differentiation, landscape genetics, subterranean rodents
Abstract:	1. During the last decade convolutional neural networks (CNNs) have revolutionized the application of deep learning methods to classification tasks and object recognition. These procedures can capture key features of image data that are not easily visible to the human eye and use them to classify and predict outcomes with exceptional precision. 2. Here we show for the first time that CNNs provide highly accurate predictions for small-scale genetic differentiation and diversity in *Ctenomys australis*, a subterranean rodent from central Argentina. Using microsatellite genotypes and high-resolution satellite imagery, we trained a simple CNN to predict local Fst and mean allele richness. To identify landscape features with high impact on predicted values we applied species distribution models to obtain the distribution of suitable habitat. Subsequent use of a Machine Learning algorithm (Random Forest) allowed us to identify the attributes that contribute the most to predictions of population genetic metrics. 3. Predictions obtained from the CNN accounted for more than

	98\pcnt~of the variation observed both in Fst and mean allele richness values. Random Forest on landscape metrics indicated that features involving connectivity and consistent prevalence of suitable habitat promoted genetic diversity and reduced genetic differentiation in *C. australis*. 4. Validation with synthetic data via simulations of genetic differentiation based on the landscape structure of the study area and of a nearby area showed that deep learning models are able to capture complex relationships between actual data and synthetic data in the same landscape and between synthetic data generated under different landscapes. 5. Our approach represents an objective and powerful approach to landscape genetics because it can extract information from patterns that are not easily identified by humans. Spatial predictions from the CNN may assist in the identification of areas of interest for biodiversity conservation and management of populations.

Deep Learning and Satellite Imagery predict genetic diversity and differentiation

Marcelo J. Kittlein[1]*, Matías S. Mora[1], Fernando J. Mapelli[2], Ailin Austrich[1] and Oscar E. Gaggiotti[3]

[1]Departamento de Biología. Instituto de Investigaciones Marinas y Costeras (IIMyC). Facultad de Ciencias Exáctas y Naturales Universidad Nacional de Mar del Plata. Consejo Nacional de Investigaciones Científica y Técnicas (CONICET), Mar del Plata, Argentina.

[2]Grupo de Genética y Ecología en Conservación y Biodiversidad (GECOBI), División Mastozoología. Museo Argentino de Ciencias Naturales "Bernardino Rivadavia", (CONICET), Ciudad de Buenos Aires, Argentina.

[3]Centre for Biological Diversity, University of St Andrews, Fife, UK.

One sentence summary

Convolutional neural networks provided highly accurate predictions of genetic structure from high resolution satellite imagery.

Abstract

1. During the last decade convolutional neural networks (CNNs) have revolutionized the application of deep learning methods to classification tasks and object recognition. These procedures can capture key features of image data that are not easily visible to the human eye and use them to classify and predict outcomes with exceptional precision.

2. Here we show for the first time that CNNs provide highly accurate predictions for small-scale genetic differentiation and diversity in *Ctenomys austaralis*, a subterranean rodent from central Argentina. Using microsatellite genotypes and high-resolution satellite imagery, we trained a simple CNN to predict local F_{ST} and mean allele richness. To identify landscape features with high impact on predicted values we applied

*Corresponding author: M. J. Kittlein kittlein@mdp.edu.ar

species distribution models to obtain the distribution of suitable habitat. Subsequent use of a Machine Learning algorithm (Random Forest) allowed us to identify the attributes that contribute the most to predictions of population genetic metrics.

3. Predictions obtained from the CNN accounted for more than 98% of the variation observed both in F_{ST} and mean allele richness values. Random Forest on landscape metrics indicated that features involving connectivity and consistent prevalence of suitable habitat promoted genetic diversity and reduced genetic differentiation in *C. australis*.

4. Validation with synthetic data via simulations of genetic differentiation based on the landscape structure of the study area and of a nearby area showed that deep learning models are able to capture complex relationships between actual data and synthetic data in the same landscape and between synthetic data generated under different landscapes.

5. Our approach represents an objective and powerful approach to landscape genetics because it can extract information from patterns that are not easily identified by humans. Spatial predictions from the CNN may assist in the identification of areas of interest for biodiversity conservation and management of populations.

Keywords— biodiversity prediction, convolutional neural networks, coastal dunes, *Ctenomys australis*, Deep Learning, genetic differentiation, landscape genetics, subterranean rodents.

Introduction

Landscape Genetics aims to explain how landscape features mold and constrain the flow of genes and individuals (Holderegger and Wagner, 2008). For this, the landscape is characterised using various spatial summaries which are then incorporated into statistical or mechanistic models aimed at explaining or predicting population genetic summaries of groups of individuals (Storfer et al., 2007). However, the possibility of obtaining highly precise predictions of population genetic traits based solely on physical attributes of the landscape where individuals live has not yet been achieved. Recent applications of Deep Learning (DL) algorithms to problems in the fields of Ecology and Evolutionary Biology

(Schrider and Kern, 2018; Ham et al., 2019) suggest that this should now be possible.

The spatial configuration of landscapes is one of the main drivers of the movement of individuals and, therefore, landscape ecologists and geneticists attempt to relate different characterisations of the landscape to the genetic composition of populations or individuals grouped under varying criteria. The standard approach is to use predefined and, to a large extent, subjective quantitative landscape descriptors such as friction or resistance maps that somehow express levels of fragmentation or availability of the species' suitable habitat (Peterson et al., 2019; Mapelli et al., 2020). Here we show for the first time that highly precise predictions of genetic diversity and differentiation can be achieved using simple DL algorithm based solely on a high-resolution satellite snapshot of the landscape (i.e. without the use of any predefined landscape descriptor).

Recent developments in DL have made complex algorithms readily available to researchers from various disciplines interested in using neural networks to predict a large variety of phenomena. Convolutional Neural Networks (CNNs), have shown great predictive power in classification and pattern recognition using image data (Gu et al., 2018). In CNNs, pixels which are close together are treated differently than pixels that are far apart, which captures fine details of the way in which a species habitat is spatially structured (i.e. with great potential to summarise landscape features that may impact spatial structuring of genetic variation). It has been used successfully in population genetics to identify selective sweeps, inferring demographic history and recombination rates (Sheehan and Song, 2016; Schrider and Kern, 2018); and also in meteorology/oceanography to predict the occurrence of ENSO events (Ham et al., 2019).

Predictions of spatial genetic structure from satellite images can be an invaluable tool for conservation of endangered species. For example, they can help identify geographic areas that are optimal for species reintroductions. However, DL cannot be easily used to identify landscape attributes that may influence population genetic structure. Nevertheless, other Machine Learning methods, such as Random Forests, can be used to obtain this information.

In what follows, we first illustrate the power of a CNN approach to predict spatial genetic diversity patterns using a small subterranean rodent as a case study. We then use

Random Forest to identify the landscape features that drive the observed patterns. In doing so we can literally say that we are 'putting the landscape in landscape genetics' (c.f. Storfer et al., 2007).

Materials and methods

We use DL algorithms to predict genetic diversity and differentiation in *Ctenomys australis*; a narrowly distributed and endangered South American subterranean mammal. Subterranean rodents of the genus *Ctenomys* (commonly known as tuco-tucos) comprise one of the most diverse genera of mammals worldwide. Over 60 species are distributed throughout the southern cone of south America (Fig. 1A) and many are threatened (Bidau, 2005; Mapelli et al., 2020). They occupy sandy soils in rather simple landscapes where they build burrow systems that are used for shelter, dwelling, and access to food. With low dispersal abilities, populations and groups of individuals attain substantial levels of genetic differentiation even between nearby locations (Kittlein and Gaggiotti, 2008; Mora et al., 2010). *Ctenomys* species, and particularly *C. australis*, are ideal for illustrating how genetic variation can be structured at very small spatial scales. They are restricted to burrows in a rather simple landscape and do not move often (Fig. 1B). All these characteristics make them ideal targets of landscape genetic studies.

We assessed the ability of a CNN to predict spatial variation of genetic diversity and differentiation of *C. australis* in a 3,000 ha tract of a coastal dune landscape in southeastern Buenos Aires province, Argentina (Fig. 1C).

The collection of samples for genetic analyses was carried out in Las Grutas ($\sim$ 10 km south of the city of Necochea, province of Buenos Aires, Argentina; 38^o 37'S - 58^o50'W), between March 2003 and April 2005. A total of 112 individuals of the herbivorous subterranean rodent *Ctenomys australis* were live-trapped and a finger snip sample taken and preserved. All individual were released back to their burrows at the point of capture. For each sample individual the multilocus genotype at nine microsatellite loci was obtained as described in Mora et al. (2010).

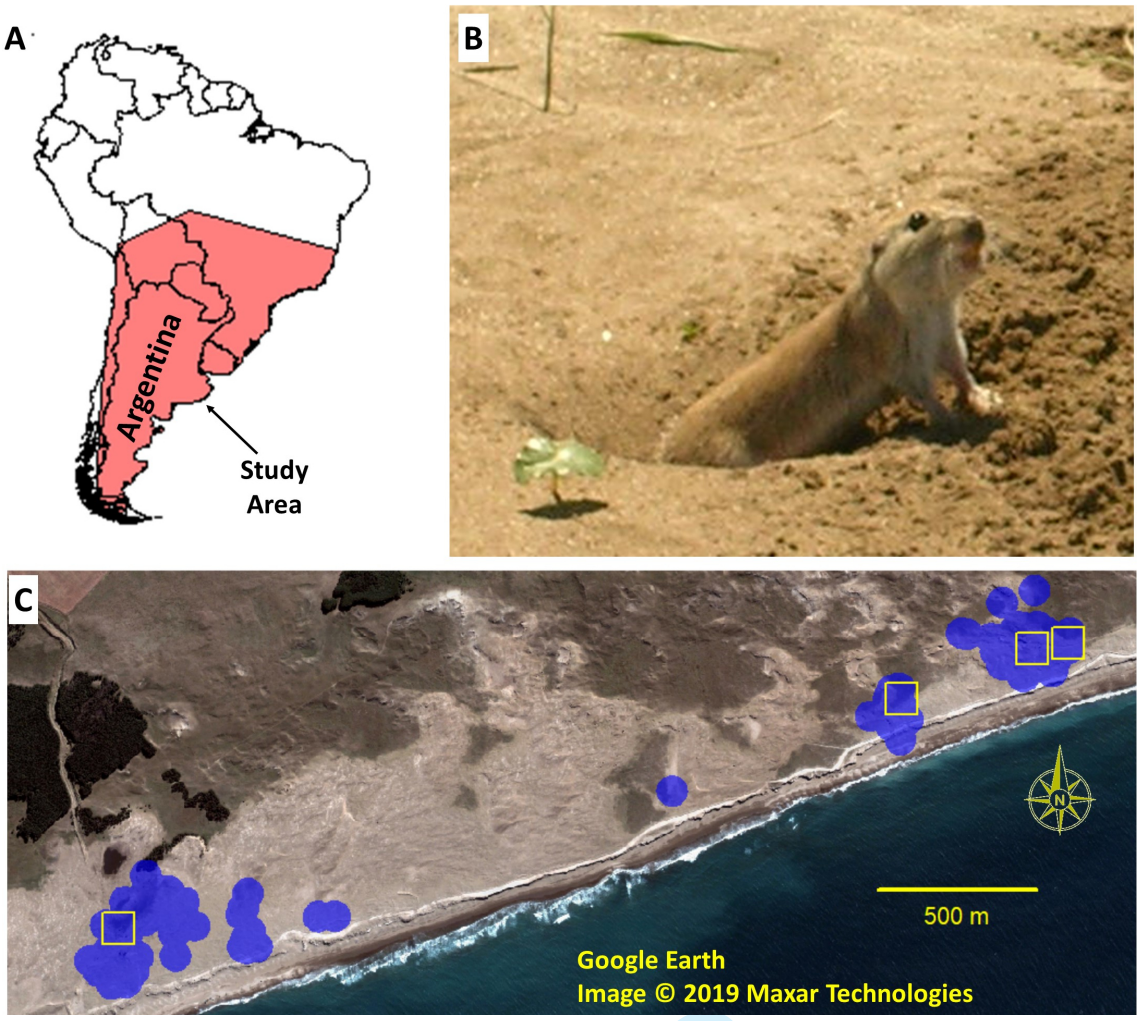

Figure 1: **Southern South America is home to one of the most diverse genera of mammals. More than 60 rodent species of *Ctenomys*, known as tuco-tucos, extend from southern Peru to Tierra del Fuego, occurying across diverse types of sandy soils where they dig their underground burrow systems.** (A) The distribution of tuco-tucos comprises the southernmost portion of South America (red polygon). Sampling of individuals was carried out $\sim$ 10 km south of Necochea, Buenos Aires province, Argentina, in a 3.5 by 1.5 km stretch of coastal dunes. (B) A juvenile female of the herbivorous subterranean rodent *Ctenomys australis* at the entance of its burrow system (photo credit M. Mora). (C) Quickbird satellite image of the study area (Latitude: -38.63466, -38.62104; Longitude: -58.87851 , -58.83692). Multilocus genotypes of 112 tuco-tucos with active home-ranges (blue shadow) and one-hectare image sections (4 randomly sampled yellow squares are shown) were used to integrate environmental and population-genetic data.

Collection and pre-processing of environmental and genetic data

The first step in our study was to randomly sample 1,000 one-hectare image sections of the landscape inhabited by *C. australis*. Two rgb image crops of the coastal landscape

including the sampling area (3.5 x 1.5 km, approx.) were downloaded in jpg format from GoogleEarth. The images correspond to a pansharpened Quickbird image of 2005-04-17 which includes the area where the genetic samples were collected and a nearby area used two validate the deep learning models. We defined 2 kml rectangles at Latitude: -38.63466 , -38.62104; Longitude: -58.87851 , -58.83692 and at Latitude: -38.66273, -38.64926; Longitude: -58.98031, -58.93872 and then downloaded the historical jpg image for that date from google earth at the maximum resolution available (Table 1). We then cropped the jpg image to the inner edges of the polygon lines and georeferenced them in R using the `raster` (Hijmans et al., 2015), `sp` (Pebesma et al., 2012), and `rgdal` (Bivand et al., 2015) packages: obtaining the rgb bands with a spatial resolution of 0.85 m.

Table 1: Technical specification for satellite imagery used in this paper.

DigitalGlobe 2005-04-17 Catalog ID: 10100100042DA105 Cloud Cover: 0%, Quality: 99
QB02 2005-04-17 0.0% 7.9^o Image ID: 10100100042DA100 Image Clouds: 0.0% Image Off Nadir: 9.6^o Bands: 4-BANDS Max GSD: 0.63m Sun Elevation: 35.1^o Max Target Azimuth: 353.7^o

Most tuco-tucos, and particularly *C. australis*, are solitary. Mean home-range size for *C. australis* is $\sim$ 1300 square-meters (Cutrera et al., 2010). A layer delimiting a radius of 50 m around each capture point was used to sample image information. Random locations within this layer were sampled and used as the center of one-hectare square polygons including at least 6 individuals (iterated randomly until this condition was satisfied). In this way, one-thousand one-hectare square polygons were randomly generated using a script written in R.

Using the genotypes of the individuals included in each polygon we calculated the mean allele diversity (MAlleles) per individual in each polygon and the genetic differentiation between each polygon and all other polygons (local-F_{ST}). For this, we used the

`basic.stats()` and `allelic.richness()` functions of the R package `Hierfstat` (Goudet et al., 2015). Associated one-hectare sections of the landscape rgb image were stored in a `csv` file with the genetic data for training a convolutional neural network. Each one-hectare section was stored as a 3 channel array image of 117 by 117 pixels.

Testing for Isolation by Distance patterns

We first evaluated whether or not the spatial population genetics structure of *C. australis* could be described by a simple isolation by distance model. For this purpose, we applied a spatial principal components analysis using functions available in the R-package adegenet (sPCA; Jombart and Ahmed, 2011). In our analysis the sPCA yielded scores summarising both genetic variability and the spatial structure among focal sections of the landscape using Moran's eigenvector maps (MEMs). Spatial variation is represented by a connection network using a K-nearest neighbours approach. Global structure is evaluated by assessing if geographically close sections are genetically more similar than expected under a random spatial distribution of genetic variation. The observed test statistic (comparable to a r-square statistic, depicting associations of alleles to vectors in the matrix of global MEMs) is compared to the distribution of test statistics obtained through a Monte Carlo randomisation procedure using 9999 permutations (Jombart and Ahmed, 2011; Vonhof et al., 2016).

Convolutional neural network

To predict the spatial variation in genetic diversity and genetic differentiation in *C. australis* using image data we built a simple convolutional neural network (CNN) that consisted of 3 convolutional layers, 2 pooling layers, a flatten layer, a dense layer and a linear output, as shown in Figure 2A. A 30% dropout was used to avoid over-fitting (see below). The CNN was coded in `Python` using `keras` and `tensorflow`. The mean squared error was used as loss function for fitting the CNN. The data was split into a training set (70%) and a validation set (30%) and the fitting procedure was run for 1000 epochs (number times that the learning algorithm updates model parameters) while monitoring validation loss. To prevent over-fitting, the procedure was stopped if no improvement was made during 50

epochs.

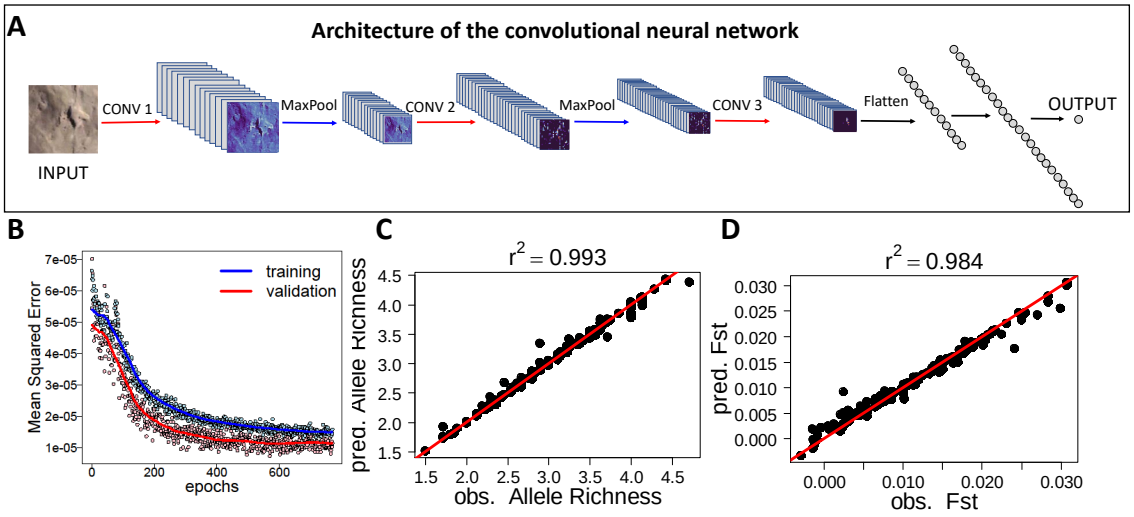

Figure 2: **Convolutional neural networks provided very accurate predictions of genetic structure and diversity at very small spatial scale in the subterranean rodent *Ctenomys australis*.** (A) The CNN consisted of 3 convolutional layers, the first two followed by a pooling layer, and the third by a flatten layer (see supplementary Material). (B) The process of training the CNN is handled by monitoring the mean squared error between the observed and the predicted statistic. Here the process is set until no improvement for the validation data subset is observed. (C) Fit of mean allelic richness. (D) Fit of genetic differentiation.

Arhictecture of the Convolutional Neural Network in Python

The CNN is simply coded with a few lines of `keras` code.

```
model = Sequential()
model.add(Conv2D(24, kernel_size = 3, activation='relu', padding='same',
    input_shape = (117, 117, 3)))
model.add(MaxPool2D())
model.add(Conv2D(48, kernel_size = 3, activation='relu', padding='same'))
model.add(MaxPool2D(padding='same'))
model.add(Conv2D(64, kernel_size = 3, padding='same', activation='relu'))
model.add(Flatten())
model.add(Dropout(0.3))
model.add(Dense(256, activation='relu'))
model.add(Dense(1, activation='linear'))
```

A CNN can be built by alternating 2D convolutional layers with other types of layers

to produce an output layer with appropriate dimensions that allow the contrast with the nature and dimension of the data to be predicted. Convolutional layers (Conv2D) process image values using filters that summarise the spatial relations of pixel values into feature maps that highlights spatial patterns from patches of pixels in image data. This process is called convolution and is implemented by matrix multiplication. Image data is processed by filters of a given size (3 x 3 filters in our example) with weight values arranged in different ways. The dimension of the filters and the values of the weights highlight spatial patterns present in image data. The output is scaled by the type of activation used (in our code above, the relu activation outputs only positive values).

In our example the input is a 1-ha section of the study area satellite image (117 x 117 pixels in red green and blue channels) that is processed yielding 24, 48 and 64 feature maps that results from filters of 3 x 3 pixels, alternating with a pooled layer by the maximum value of each filter (results in downsampled or pooled feature maps that highlight the most present feature). A flatten layer then arranges the feature maps to a single dimensional layer that is passed to the fully connected layer (Dense) after putting 30% of its values to zero (Dropout layer that regularises the process reducing overfitting). The final Dense layer has one output value for each input that is contrasted against observations to fit the model. A good description with instructive visualisations can be reached at https://methodsblog.com/2019/11/13/understanding-deep-learning/; for a recent reference in this area see Norouzzadeh et al. (2021).

CNN Validation

The standard procedure to validate a neural network is to determine how accurate the trained model predicts the validation set, which in our case consists of 300 images and their corresponding population genetics summary statistics (mean allele diversity and local F_{ST}). However, in this study we carried out two additional validation procedures consisting on the use of augmented data and a simulation study where synthetic data were obtained by first generating cost-distance matrices from the satellite image tracts and then using a landscape genetic simulator to generate synthetic genetic data. We explain these two additional strategies below.

Data augmentation

To demonstrate that the CNN can identify relevant landscape features regardless of their orientation or position, we applied data augmentation to the image training data randomly shifting up to 30% of the image both vertically and horizontally , and also rotating randomly up to 30 degrees. This provides a new set of images that we used to train the CNN (no original image was used for training). We then applied the trained model to the validation set consisting of images that were not used for data augmentation and evaluated the ability of the CNN to predict observed genetic structure. We used the `ImageDataGenerator()` function in `keras`. Fig. 3 shows an example of this procedure.

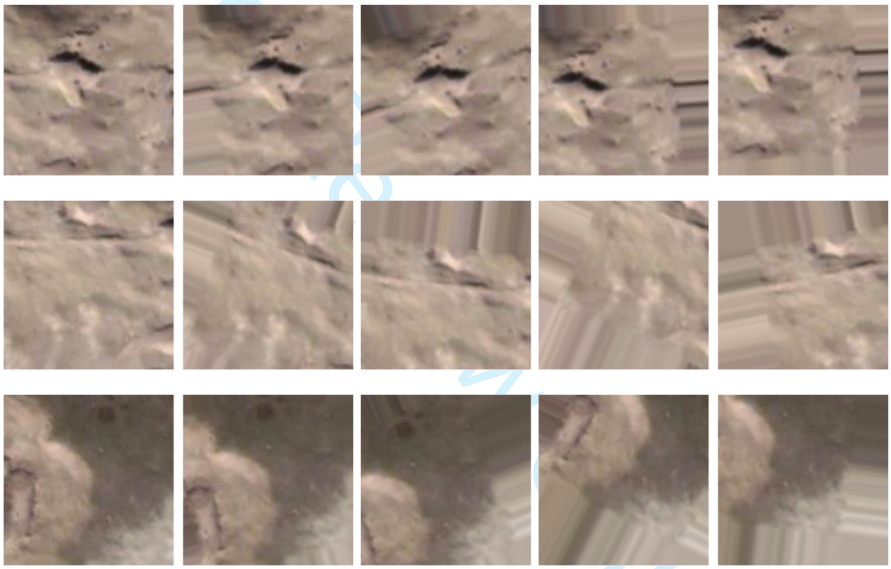

Figure 3: Augmented images. The first image in each row shows the original image. The 4 additional images of each rows show augmented images that were generated by randomly shifting and rotating the original image.

Simulation Study

The standard procedure to validate statistical methods in population genetics is to apply the method to synthetic data generated using a simulation program. In our case, this involves generating genetic data for each landscape section and then evaluating the accuracy of the CNN model predictions of population genetic structure. No population genetic simulator can take satellite images as input so we first had to transform the images into landscape data describing habitat suitability and physical connectivity between habitat

patches. This necessarily leads to a substantial loss of the landscape information contained in an image, which in turn leads to the generation of synthetic genetic data that do not fully correspond to the satellite image. Nevertheless, some of the image information should be transferred to the input file describing landscape attributes and should drive the spatial patterns observed in the simulated genetic data.

The synthetic genetic data was simulated using the program CDPOP (Landguth and Cushman, 2010). CDPOP is a spatially explicit, individual-based landscape genetic model, where the processes of birth, death, dispersal and mating of individuals can be simulated in complex landscapes through time. The program models genetic exchange among individuals as probabilistic functions of movement cost among individuals' locations. These probabilities can be approximated from a layer of resistance to movement between the spatial coordinates of individuals based on characteristics of the habitat where the species lives.

In a first step, we used the occurrence coordinates of tuco-tucos and the layer of suitable habitat obtained with maxent (Phillips and Dudík, 2008, see below) to approximate the probabilities of individuals' movement between habitat patches. The spatial layer of suitability values was transformed into a layer of movement cost using the the mean suitability value of the 16 neighbours of focal pixels (R package gDistance; van Etten, 2017). Then we calculated a matrix of least cost path distances between the coordinates of individuals using the function shortestPath() and lineLength() in package SDraw (McDonald and McDonald, 2020). This procedure yields a distance matrix similar to a matrix of geographic distances, except that for nearby points separated by patches of unsuitable habitat, pairwise distances are increased because the paths between these locations circumvent unsuitable patches (Figure S2). This cost-distance matrix is used by CDPOP to calculate the probability of gene flow between locations using movement functions to describe how the probability decreases with distance. We used a negative exponential ($p = A \times 10^{-B \times d}$; with A=1 and B=0.01, which reduces probability [p] to extremely low values for cost distances [d] above 500 m). To model the temporal evolution of gene flow with CDPOP, we chose demographic and genetic parameters to match as closely as possible known features of the population biology of *Ctenomys australis* (see supplementary Material Table S1).

The first 20 generations of each replicate run did not allow for gene flow to let the demographics settle down. These were followed by 40 generations with gene flow during which we monitored changes in F_{ST} through time. The multilocus genotypes where sampled one generation after the local-F_{ST} reached observed values in the real dataset. We predicted genetic differentiation to this new set of genetic data using the CNN model fitted with real genotype data. We did not expect these predictions to have a close match with synthetic data; firstly because it is likely that the process of gene flow modelled with CDPOP does not mimic all the complexities of the actual recent microevolution of the species in its landscape habitat; and secondly because the actual gene flow between tuco-tucos' habitat patches is likely influenced by other landscape features not considered in the friction layer used to obtain the synthetic genetic data. Neither the complexities of the process nor the constrains imposed by the landscape are fully incorporated in the simulations.

We employed two different validation strategies. In the first one, the CDPOP simulations were based on the above-described maxent approximation to the landscape structure of our study area (Area 1 in Figure S1), and considered a total of 112 individuals and 9 loci. Individuals were placed in the actual geographic coordinates. We sampled the 112 genotypes and then evaluated the accuracy of the CNN model trained on the real genetic data and satellite image to predict the genetic structure of this simulated data.

The second type of validation was based on the landscape structure of our study area, but now considered a total of 1000 individuals and 100 loci. Individuals were randomly placed in the landscape using suitability values as probabilities of occurrence. This yielded a spatial distribution resembling that of tuco-tucos; with suitable patches having many clustered individuals and with a few scattered individuals located in less suitable areas. We then obtained the cost distance matrix and simulated gene flow as described above. To generate a dataset similar to the real one, we obtained a sample of individuals and loci that matched the numbers in our actual samples. We trained the CNN with this synthetic dataset and then transferred what the CNN learned to a nearby area for which we didn't have genetic samples (8 km to the west of our study site, Area 2 in Figure S1). We used the same procedure as before to derive a movement cost layer for CDPOP that corresponded to this new area and run CDPOP simulations with the same parameter values and sampling

procedure as described above to obtain a new synthetic dataset and then predicted F_{st} values with the CNN trained with synthetic data generated for the study area.

We use local-F_{ST} as a validation metric since we believe the conditions that limit the dispersal of individuals and genetic exchange can be more directly approximated through the distribution of habitat suitability and that its signature is better represented in satellite imagery than metrics related to genetic diversity. At this stage, we did not identify any straightforward link with genetic diversity that can be easily quantified in the information available in satellite imagery. We recognise, however, that there are many other subtle details of the landscape that cost-distance matrices do not incorporate and that surely have unaccounted effects on local isolation and influenced gene flow between neighbouring sites.

Distribution of suitable habitat, landscape metrics and Random Forest model

Although deep learning methods have great predictive power, it is difficult to accurately identifying which variables or traits are the drivers of their great predictive capabilities. To overcome this limitation, we estimated the distribution of suitable habitat for tuco-tucos in the area using species distribution models with Maxent (Phillips et al., 2017). We then summarised the spatial configuration of patches of suitable habitat with a set of 65 landscape metrics (that measure variation in shape, area, connectivity, etc.; see Table S2) and selected a subset that was free of multicollinearity to carry out a Random Forest analysis.

We used the occurrence coordinates of tuco-tucos and the 3-channel-image to estimate the distribution of suitable habitat in the area using the function `maxent()` in the R-package `dismo` (Hijmans et al., 2017). A binary layer of suitable habitat was obtained using the kappa statistic as threshold to set apart pixels of suitable habitat (Hijmans et al., 2017).

Using the layer of suitable habitat, we calculated landscape metrics for each one-hectare square using functions in package `landscapemetrics` (Hesselbarth et al., 2019, see Table S2). To simplify the interpretation of the analysis and avoid redundancy in the set of landscape metrics used as predictors of the genetic indexes, we removed those variables with a linear correlation above 0.8, using the function `findCorrelation()` in the R package

311 `caret`.

312 The resulting uncorrelated subset included the following landscape metrics: `area_mn`,
313 `area_sd`, `cai_cv`, `circle_mn`, `circle_sd`, `cohesion`, `contig_cv`, `contig_sd`, `dcad` ,
314 `dcore_cv`, `enn_cv`, `enn_mn`, `frac_sd`, `gyrate_cv`, `lpi` , `mutinf`, `pafrac`, `shape_cv`,
315 `shape_mn`.

316 We fitted a Random Forest model using the landscape metrics as predictors and F_{ST}
317 or MAlleles as response. A subset of 30% of the data was left out for validation. The
318 function `RandomForestRegressor()` from the python package `scikit-learn` was used to
319 fit the model using the mean squared error between the predicted and the observed response
320 as loss function.

321 Random Forest models allow for the evaluation of the way in which predictors con-
322 tribute to the prediction of the response variable. We used Shapley values to estimate the
323 relative contribution of predictors and to visualize relationships for model interpretation
324 (García and Aznarte, 2020). The functions `shap.TreeExplainer()` and `shap.summary_plot()`
325 from the phyton package `shap` (see Lundberg et al., 2020) were used to calculate and plot
326 shapley values.

327 Results

328 The distribution of focal F_{ST} values, depicting small-scale genetic differentiation in tuco-
329 tucos (Fig. 4A), reflects the low mobility of these animals. F_{ST} values observed for other
330 mammalian species at spatial scale of several square kilometres (mean±sd 0.11±0.008;
331 Lawrence et al., 2019), are recorded on a spatial scale of only a few hectares (Fig. 4A).
332 The results of the isolation by distance analysis showed that distance cannot account for
333 genetic differentiation in tuco-tucos (tests for global structure using Moran's eigenvector
334 maps; Fig. 4B; r^2=0.26; P=0.22). Genetic diversity, expressed by the mean number of
335 alleles per individual was rather low as compared with other mammalian species (mean±sd
336 3.08±0.72; Lawrence et al., 2019).

337 The proportion of the variation in mean allele richness and genetic differentiation (F_{ST})
338 explained by the CNN were both surprisingly high, above 98% (Fig. 2C-D), an unexpected
339 outcome that suggests great promise for the application of these algorithms in landscape

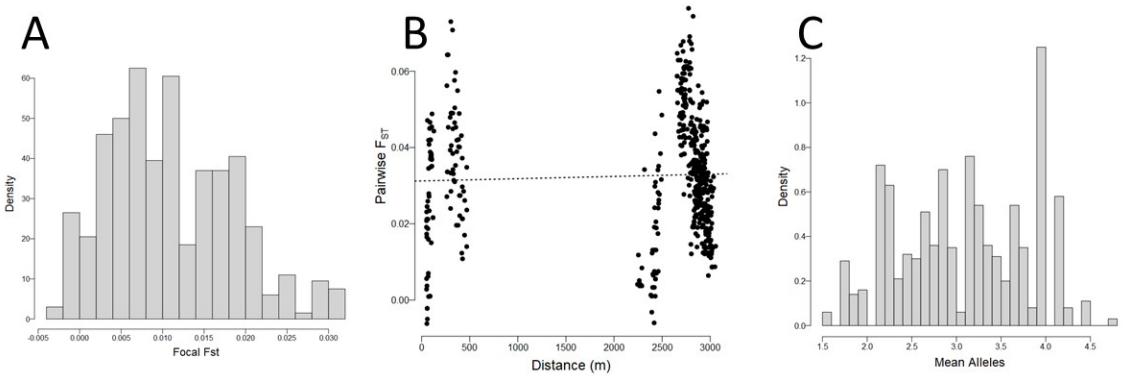

Figure 4: A. Distribution of focal F_{ST} values in the study area. B. Isolation by distance of pairwise F_{ST} values; regression relationship is shown by the dotted line. C. Distribution of mean allele number per individual.

genetics. To evaluate to which extent the CNN could be used to predict outcomes in a landscape that differs from that on which it is trained, we next trained the CNN using 1ha sections that were modified (augmented) by rotating and displacing the original images and validated it with the remaining 30% 1ha images used in the first analysis. Even with augmented data the r^2 was ~ 0.8 (see Supplementary material).

Some straightforward transfers of the fitted CNN are the ability to extrapolate to other ranges beyond those used for training and validation of the model (Fig. 5). In the case of the sand dune tuco-tuco, continuous areas with low plant cover support greater mean allelic richness (Fig. 5A) and transitional areas from low to high plant cover or tree plantations are more genetically differentiated (Fig. 5B).

The ability of the CNN to predict genetic differentiation with synthetic data was relatively good and rather unexpected when transferring what the CNN learned with actual data to synthetic data in the same landscape context. Here, using the same landscape scenario but with synthetic genetic data brought about from a demographic process that surely simplifies actual microevolutionary dynamics of tuco-tucos we obtained an unexpected close match (r^2=0.61, Fig. 6A). A close match was also obtained when transferring the same demography to a different landscape context. Here, after fitting a new CNN model to simulated data in the study area, we predicted genetic differentiation to simulated data in this new area using image information previously unknown to the CNN (r^2=0.57, Fig. 6B).

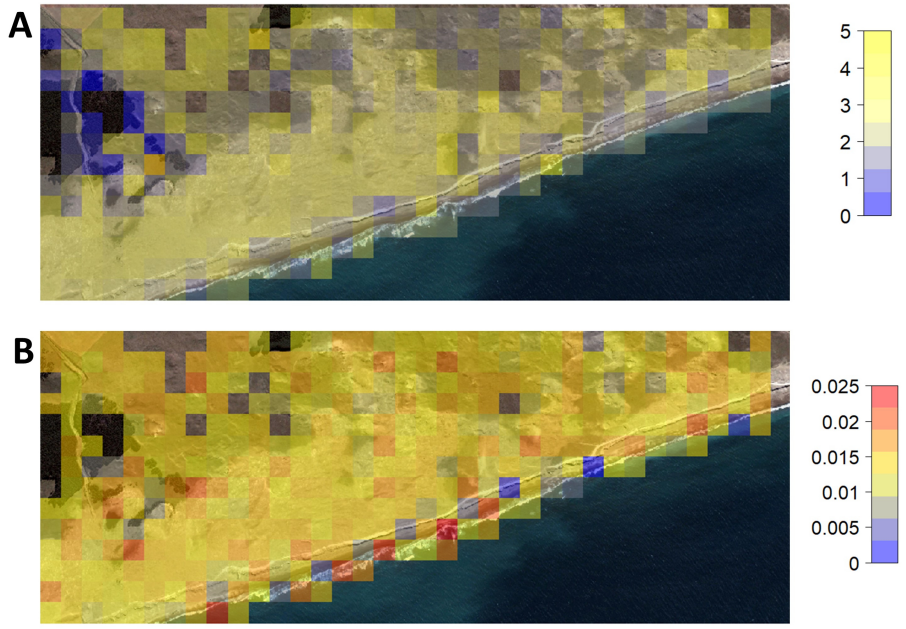

Figure 5: **Convolutional neural networks are able to integrate complex features of the lanscape for predicting population genetic characteristics**. (A) Spatial prediction of Allele richness (higher richness along continuous sparsely vegetated dune habitats is observed). (B) Spatial prediction of genetic differentiation (greater differentiation at the border of the landscape or at transitions between habitat types is observed).

As stated before, despite their predictive power, DL methods cannot clearly identify the specific landscape attributes that drive genetic structuring of natural populations. Thus, we used a different machine learning approach, Random Forests, to identify landscape attributes underlying observed spatial genetic diversity patterns (see above). The results showed great prediction accuracy, with above 96% of variation explained for genetic diversity and differentiation in *C. australis*. Although not having as high a prediction accuracy as that of CNNs, RF models are able to provide insight into which and to what extent traits/variables contribute to the prediction (via Shapley values, Nandlall and Millard, 2019; García and Aznarte, 2020, Fig. 7).

Predicted values for mean allele richness were mainly driven by three landscape features (Fig. 7C). Variation in patch shape (Fig. 7D) and mean distance to neighbor patches (Fig. 7E) impacted negatively upon mean allelic richness, while patch cohesion had a positive impact (Fig. 7F). Overall, those features involving connectivity and consistent prevalence of suitable habitat were associated to higher genetic diversity. The predictive power of landscape metrics on genetic differentiation was less clear, with many landscape

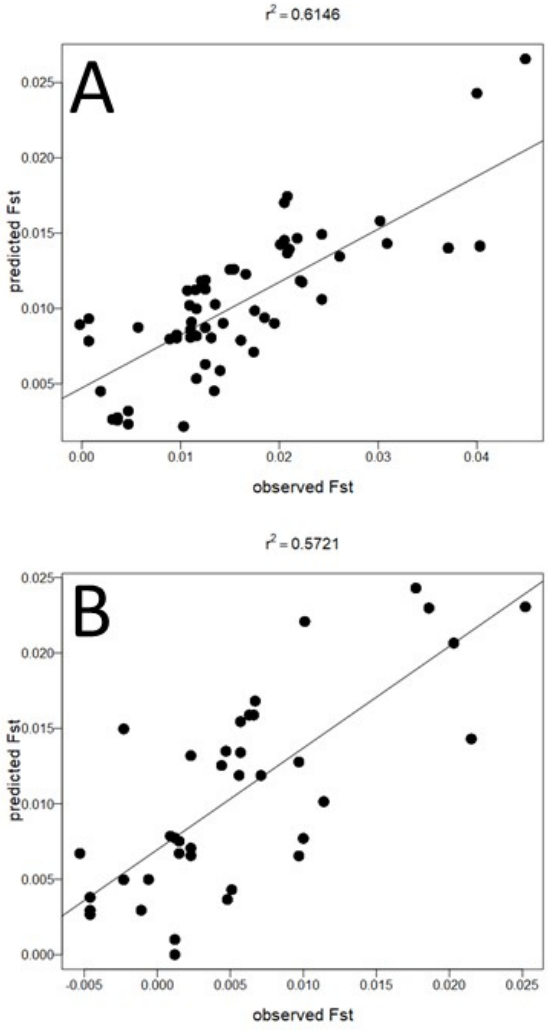

Figure 6: A. Predictions made by the CNN model fitted with actual data to Fst values from synthetic data obtained with CDPOP using the same coordinates of tuco-tucos sampled in the study area and demographic parameters trying to depict as close as possible those of *Ctenomys australis*. B. Predictions of synthetic Fst values from synthetic data obtained with CDPOP in a new area $\sim$ 8 km west of the study area from a CNN model fitted to synthetic data generated using CDPOP in the landscape of the study area.

features having a limited contribution to predicted F_{ST} values (Fig. 7G). The two most important metrics describe the compactness of the patches within a 1ha plot in terms of its mean and standard deviation (Figs. 7H and 7I). The less compact the habitat patches are and the lower the variation in compactness across them, the stronger the local F_{ST} because the individuals are likely to be more isolated, increasing the strength of genetic drift in that particular landscape section. The third most important predictor is the variation in nearest neighbour distance across patches (Fig. 7J), with increasing local F_{ST} as this variation increases. Therefore,the higher the variation in habitat patch connectivity within

383 a landscape section, the stronger the effect of genetic drift.

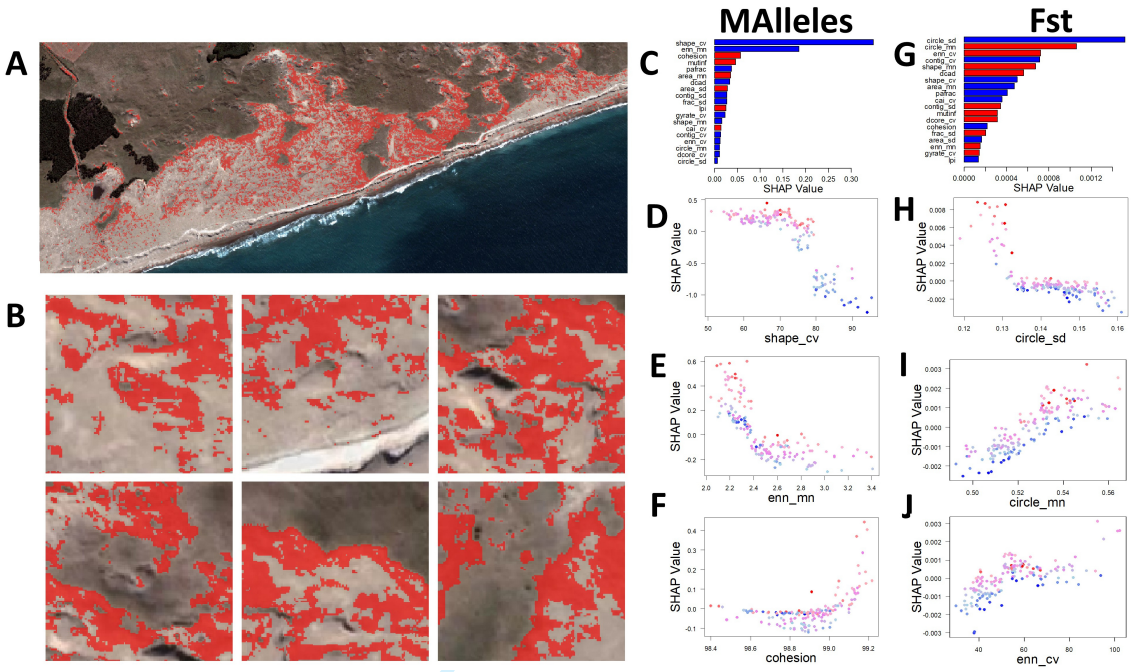

Figure 7: **The spatial configuration of suitable habitat as summarized by several landscape metrics were used to interpret how landscape features impact on genetic diversity and differentiation in the subterranean rodent *Ctenomys australis*.** (A) Estimation of the spatial arrangement of suitable habitat (red) derived using Maxent. (B) A subset of one-hectare sections of coastal dune landscape showing the distribution of patches of suitable habitat (red). Sections are characterized by different landscape metrics (see Supplementary material) having varying impact on genetic diversity and differentiation. (C) Random Forest model for mean allele richness and feature importance of landscape metrics with positive (red) and negative (blue) impact on the predicted response. (D to F) Relationship between the tree most important landscape features and SHAP values in Random Forest models for Mean Allele diversity. (G) Random Forest model for genetic differentiation (Fst) and Feature importance of landscape metrics with positive (red) and negative (blue) impact on the response. (H to J) Relationship between the three most important landscape features and SHAP values in Random Forest model for genetic differentiation (Fst). The color of the points in D–F and H–J represents the variation in the scale of the response variable from low values (in blue) to high values (in red).

384 Discussion

385 This study provides a blueprint for the application of Machine Learning (ML) techniques
386 and satellite imagery in landscape genetics. Our approach includes several steps, which
387 also involve well established landscape ecology approaches such as Species Distribution
388 Models. Briefly, we used satellite imagery and population genetics data to train a CNN

that predicts spatial patterns of genetic variation with great accuracy. Then, in order to identify the landscape features driving this high predictive power, we applied SDM to obtain maps with the spatial arrangement of suitable habitat and used landscape metrics to summarise their landscape-level attributes. Subsequent use of Random Forest regression allowed us to identify those attributes that best explained the observed spatial patterns. The two machine learning techniques we used are complementary; CNNs allow us to predict the spatial genetic structure that a particular landscape could maintain simply from a satellite image. On the other hand, RFs can uncover complex non-linear effects of particular landscape features on the structuring of genetic variation.

Our results indicate that this new combined ML strategy represents an more objective and powerful approach to landscape genetics than the traditional use of predefined and fairly subjective environmental friction maps. Moreover, they also lead to much higher predictive accuracy as they can extract information from patterns that are not easily identified by humans. In the case of CNNs, all the information contained in a landscape is used to predict genetic structuring while in the case of RFs, we can use an arbitrarily large number of metrics that may interact in nonlinear ways to infer the mechanisms responsible for the observed spatial patterns of genetic variation.

We used three different strategies to evaluate the accuracy of CNNs for prediction of spatial genetic structure. The first one consists of the standard approach used in Machine Learning, namely the use of a validation set consisting in our case of satellite images and the observed values of population genetics summary statistics corresponding to the area covered by the image, which are different from the training data. Another strategy was to use data augmentation to generate modified images of the studied area in order to change the orientation and location of the landscape features contained in the images to demonstrate that the high predictive power of CNNs was not due to overfitting. Finally, we also implemented a validation strategy based on simulations, which is traditionally used in population genetics to validate statistical methods. Implementing this last approach was very challenging because there are no simulators that can take satellite images as input. Therefore, we used niche modelling to generate habitat suitability maps from which we obtained cost distance matrices that were used by the CDPOP to carry out individual

based simulations. This procedure necessarily leads to a substantial loss of the information present in the satellite images. The fact that, although still high, the predictive power of the CNN using this validation method was lower than that of the two other validation approaches suggests that friction or cost-distance maps do not completely capture the details of how some particular features of the landscape drive population genetic structure. This in turn highlights the need for further development of simulation techniques to incorporate the subtle details of the landscape that a satellite image contains. This would provide not only a tool for more comprehensible validations of deep learning approaches for population genetics applications but also for shedding new light on mechanistic details of how a species perceives its habitat. Some additions to recent software developments could contribute to complete this toolbox in the near future (Rebaudo et al., 2013; Terasaki Hart et al., 2021).

Importantly, our approach could be used to address pressing problems in conservation biology, such as identifying appropriate areas for endangered species re-introduction efforts. A very recent habitat suitability analysis to inform species re-introductions propose the combination of SDM approaches based on anthropogenic and ecological variables and population density estimates (Martínez-Meyer et al., 2021) that provide a proxy for habitat quality. Here, we propose that spatial genetic structure predicted from satellite imagery by Deep Learning represents a very valuable measure of habitat suitability for re-introductions because areas that can sustain high genetic diversity are likely to also improve a species resilience to future perturbations. It can also prove extremely useful for the management of wild species in habitats impacted by human activities.

The species we chose to exemplify the power of DL for landscape genetics studies has a very limited dispersal ability so we could focus on very small-scale patterns in genetic variation. Species with higher dispersal ability will need to be studied at larger spatial scales but the power of DL approaches should remain the same. Further developments in machine learning will eventually allow a better interpretation of DL algorithms to quantify the way they perceive nature (Azodi et al., 2020) so we expect rapid progress in the adoption of machine learning, and DL in particular, by molecular ecologists and conservation scientists.

Acknowledgments

To all members of Laboratorio de Ecofisologia (IIMyC-CONICET. FCEyN. UNMdP) and Grupo de Genética y Ecología en Conservación y Biodiversidad (GECoBi, Museo Argentino de Ciencias Naturales "Bernardino Rivadavia") for their invaluable support and advice. To Kaggle community members, specially to Chris Deotte, for generously sharing ideas, procedures and insight about Deep Learning. We thank Erin Landguth for her comments that greatly improved this manuscript and for her guidance on the use of CDPOP to validate our analyses. Comments by an anonymous reviewer also greatly helped improve the original manuscript. Financial support was provided by Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET, PIP 11220150100066 CO), UNMdP (Project EXA903 /18) and FONCYT (PICT 201-0427).

Author contributions

MJK, MSM and FJM designed the study. MJK designed and assembled the CNN model and code, and OEG and AA devised and formalised Random Forest and landscape metric feature analysis. All authors contributed to the final manuscript.

Competing interests

The authors declare no competing interests.

Data availability

All the data used in this study are available at ZENODO (https://doi.org/10.5281/zenodo.5535024).

References

Azodi, C. B., Tang, J., and Shiu, S.-H. (2020). Opening the black box: Interpretable machine learning for geneticists. *Trends in genetics*, 36(6):442–455.

Bidau, C. J. (2005). Family Ctenomyidae Lesson, 1842. In Patton, J. L., Pardiñas, U. F., and D'Elía, G., editors, *Mammals of South America, volume 2: rodents*, pages 818–877. University of Chicago Press.

Bivand, R., Keitt, T., Rowlingson, B., Pebesma, E., Sumner, M., Hijmans, R., Rouault, E., and Bivand, M. R. (2015). Package 'rgdal'. *Bindings for the Geospatial Data Abstraction Library. Available online: https://cran. r-project. org/web/packages/rgdal/index. html (accessed on 15 October 2017)*.

Cutrera, A. P., Mora, M. S., Antenucci, C. D., and Vassallo, A. I. (2010). Intra-and interspecific variation in home-range size in sympatric tuco-tucos, *Ctenomys australis* and *C. talarum*. *Journal of Mammalogy*, 91(6):1425–1434.

García, M. V. and Aznarte, J. L. (2020). Shapley additive explanations for no2 forecasting. *Ecological Informatics*, 56:101039.

Goudet, J., Jombart, T., and Goudet, M. J. (2015). Package 'hierfstat'. *R package version 0.04-22. Retrieved from http://www. r-project. org, http://github. com/jgx65/hierfstat*.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., et al. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377.

Ham, Y., Kim, J., and Luo, J. (2019). Deep learning for multi-year enso forecasts. *Nature*, 573(7775):568–572.

Hesselbarth, M. H., Sciaini, M., With, K. A., Wiegand, K., and Nowosad, J. (2019). landscapemetrics: an open-source r tool to calculate landscape metrics. *Ecography*, 42:1648–1657.

Hijmans, R. J., Phillips, S., Leathwick, J., and Elith, J. (2017). *dismo: Species Distribution Modeling*. R package version 1.1-4.

Hijmans, R. J., Van Etten, J., Cheng, J., Mattiuzzi, M., Sumner, M., Greenberg, J. A., Lamigueiro, O. P., Bevan, A., Racine, E. B., Shortridge, A., et al. (2015). Package 'raster'. *R package*, 734.

Holderegger, R. and Wagner, H. H. (2008). Landscape Genetics. *BioScience*, 58(3):199–207.

Jombart, T. and Ahmed, I. (2011). adegenet 1.3-1: new tools for the analysis of genome-wide snp data. *Bioinformatics*.

Kittlein, M. J. and Gaggiotti, O. E. (2008). Interactions between environmental factors can hide isolation by distance patterns: a case study of ctenomys rionegrensis in uruguay. *Proceedings of the Royal Society B: Biological Sciences*, 275(1651):2633–2638.

Landguth, E. L. and Cushman, S. (2010). CDPOP: a spatially explicit cost distance population genetics program. *Molecular ecology resources*, 10(1):156–161.

Lawrence, E. R., Benavente, J. N., Matte, J.-M., Marin, K., Wells, Z. R., Bernos, T. A., Krasteva, N., Habrich, A., Nessel, G. A., Koumrouyan, R. A., et al. (2019). Geo-referenced population-specific microsatellite data across american continents, the macropopgen database. *Scientific data*, 6(1):1–9.

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., and Lee, S.-I. (2020). From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence*, 2(1):56–67.

Mapelli, F. J., Boston, E. S., Fameli, A., Fernández, M. J. G., Kittlein, M. J., and Mirol, P. M. (2020). Fragmenting fragments: landscape genetics of a subterranean rodent (Mammalia, Ctenomyidae) living in a human-impacted wetland. *Landscape Ecology*, 35(5):1089–1106.

Martínez-Meyer, E., González-Bernal, A., Velasco, J. A., Swetnam, T. L., González-Saucedo, Z. Y., Servín, J., López-González, C. A., Oakleaf, J. K., Liley, S., and Heffelfinger, J. R. (2021). Rangewide habitat suitability analysis for the mexican wolf (canis lupus baileyi) to identify recovery areas in its historical distribution. *Diversity and Distributions*.

McDonald, T. and McDonald, A. (2020). *SDraw: Spatially Balanced Samples of Spatial Objects*. R package version 2.1.13.

Mora, M. S., Mapelli, F. J., Gaggiotti, O. E., Kittlein, M. J., and Lessa, E. P. (2010). Dispersal and population structure at different spatial scales in the subterranean rodent Ctenomys australis. *BMC genetics*, 11(1):9.

Nandlall, S. D. and Millard, K. (2019). Quantifying the relative importance of variables and groups of variables in remote sensing classifiers using shapley values and game theory. *IEEE Geoscience and Remote Sensing Letters*, 17(1):42–46.

Norouzzadeh, M. S., Morris, D., Beery, S., Joshi, N., Jojic, N., and Clune, J. (2021). A deep active learning system for species identification and counting in camera trap images. *Methods in Ecology and Evolution*, 12(1):150–161.

Pebesma, E., Bivand, R., Pebesma, M. E., RColorBrewer, S., and Collate, A. (2012). Package 'sp'. *The Comprehensive R Archive Network*.

Peterson, E. E., Hanks, E., Hooten, M. B., Ver Hoef, J. M., and Fortin, M. (2019). Spatially structured statistical network models for landscape genetics. *Ecological Monographs*, 89(2):e01355.

Phillips, S. J., Anderson, R. P., Dudík, M., Schapire, R. E., and Blair, M. E. (2017). Opening the black box: An open-source release of maxent. *Ecography*, 40(7):887–893.

Phillips, S. J. and Dudík, M. (2008). Modeling of species distributions with maxent: new extensions and a comprehensive evaluation. *Ecography*, 31(2):161–175.

Rebaudo, F., Le Rouzic, A., Dupas, S., Silvain, J.-F., Harry, M., and Dangles, O. (2013). Simadapt: an individual-based genetic model for simulating landscape management impacts on populations. *Methods in Ecology and Evolution*, 4(6):595–600.

Schrider, D. R. and Kern, A. D. (2018). Supervised machine learning for population genetics: a new paradigm. *Trends in Genetics*, 34(4):301–312.

Sheehan, S. and Song, Y. S. (2016). Deep learning for population genetic inference. *PLoS computational biology*, 12(3):e1004845.

Storfer, A., Murphy, M., Evans, J., Goldberg, C., Robinson, S., Spear, S., Dezzani, R., Delmelle, E., Vierling, L., and Waits, L. (2007). Putting the 'landscape'in landscape genetics. *Heredity*, 98(3):128–142.

Terasaki Hart, D. E., Bishop, A. P., and Wang, I. J. (2021). Geonomics: forward-time, spatially explicit, and arbitrarily complex landscape genomic simulations. *Molecular Biology and Evolution*, 38:4634—-4646.

van Etten, J. (2017). R package gdistance: Distances and routes on geographical grids. *Journal of Statistical Software*, 76(13):21.

Vonhof, M. J., Amelon, S. K., Currie, R. R., and McCracken, G. F. (2016). Genetic structure of winter populations of the endangered indiana bat (myotis sodalis) prior to the white nose syndrome epidemic: implications for the risk of disease spread. *Conservation genetics*, 17(5):1025–1040.

Supplementary Information

Processing of genetic and image data

To integrate image and genetic data a script in R handled microstatellite data and geographic coordinates to obtain summary statitics of genetic data and the corresponding image data for one-hectare sections of the landscape. Two summary statistics were obtained, the mean number of alleles per individual (mAlleles) and a measure of genetic differentiation of the individuales included in the one-hectare sections with respect of the rest of individuals in the sample (F_{ST}). Data on the genetic summaries and associated image data were saved to a .csv file for predicting genetic data from image data using Deep Learning algorithms.

```
library(raster)
library(dismo)
library(rgdal)
library(hierfstat)
library(pegas)
library(rgeos)
library(data.table)
library(readr)
library(jpeg)


ll="+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0"
utm="+proj=utm +zone=21 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +
    towgs84=0,0,0"

img1 <- readJPEG("area.jpg")

red1=raster(255*img1[,,1])
green1=raster(255*img1[,,2])
blue1=raster(255*img1[,,3])

area=stack(list(red1, green1, blue1))
proj4string(area)=ll
extent(area)=extent(readOGR("area.kml"))


Rast=projectRaster(area, crs=utm, method="ngb")

australis=read.structure("micros.Stru", n.ind=112, n.loc=9, onerowperind=T,
     col.lab=1, col.pop=2, col.others = c(3,4), row.marknames=1, NA.char="
    -9", ask=F)

dfAus=genind2df(australis)
dfAus[,2:10]=as.integer(unlist(dfAus[,2:10]))

tucos = as.data.frame(australis$other)

colnames(tucos)=c("Long", "Lat")
tucos$Long=type.convert(as.character(tucos$Long))
tucos$Lat=type.convert(as.character(tucos$Lat))
coordinates(tucos)=c("Long", "Lat")

proj4string(tucos)=ll
tucos=spTransform(tucos, proj4string(Rast))

extP=extent(c(xmin=336501.5, xmax=340091.5,  ymin=-4277834, ymax=-4276415))
```

```
Rast=crop(Rast, y=extP)

D=distanceFromPoints(Rast, tucos)
indi=which(D[]<50)
D[]=NA
D[indi]=1
np=data.frame(randomPoints(D, n=1000))
coordinates(np)=c("x", "y")
proj4string(np)=proj4string(Rast)

for(j in 1:1000){
incluidos=1
  while(length(incluidos)<=6){
    polyG = gBuffer(np[sample(1:1000, size=1)], width=50, capStyle = "
    SQUARE" )
    incluidos=which(is.na(over(tucos, polyG))==F)
  }

img=crop(Rast, y=polyG)

dfAus$pop=2
dfAus$pop[incluidos]=1

Fst=basic.stats(dfAus)$overall["Fst"]
NumAle=sum(allelic.richness(dfAus[incluidos,])$Ar)

GeneImageData=data.frame(t(c(length(incluidos), Fst, NumAle, img[])))

write_csv(GeneImageData, "giData.csv", append = TRUE)
}
```

This script saves a csv file (giData.csv) with the number of individuals sampled in each one-hectare square, the number of alleles in each square, the F_{ST} index, and the rgb pixel values for each section of one-hectare squares of the image. The mean number of alleles per individual (mAlleles) is obtained by dividing the number of alleles by the number of individuals.

Prediction of genetic indexes from image data

To explain the spatial variation in genetic diversity and genetic differentiation in *C. australis* using image data we built a simple convolutional neural network (CNN) that consisted of 3 convolutional layers, 2 pooling layers, a flatten layer, a dense layer and a linear output. A 30% dropout was used to avoid over-fitting (see code). The CNN was coded in Python using keras and tensorflow. The mean squared error was used as loss for fitting the CNN. The data was split in a training set (70%) and a validation set (30%) and the fitting procedure was run for 1000 epochs monitoring the validation loss and recording the model at each improvement of the validation loss. To prevent over-fitting if no improvement was made during 50 epochs the procedure was stopped.

Python code for trainning the Convolutional Neural Network

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import *
from keras.utils.np_utils import to_categorical
from keras.models import Sequential, load_model
```

```
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D,
    BatchNormalization
from keras.preprocessing.image import ImageDataGenerator
from keras.callbacks import EarlyStopping, ReduceLROnPlateau,
    LearningRateScheduler, ModelCheckpoint
import matplotlib.pyplot as plt
import tensorflow as tf
from keras.optimizers import Adam

train = pd.read_csv("giData.csv", header=None)

# Fst values to Y_train
Y_train = train[[1]]

# predictors in columns 4 to 41070
X_train = train.drop(train.columns[[range(3)]], axis = 1)
X_train = X_train / 255.0
X_train = X_train.values.reshape(-1,117,117,3, order="F")

# custom R2-score metrics from keras backend
from keras import backend as K

def r2_keras(y_true, y_pred):
    SS_res =  K.sum(K.square(y_true - y_pred))
    SS_tot = K.sum(K.square(y_true - K.mean(y_true)))
    return ( 1 - SS_res/(SS_tot + K.epsilon()) )

# Buil model
model = Sequential()
model.add(Conv2D(64, kernel_size = ks, activation='relu', padding='same',
    input_shape = (117, 117, 3)))
model.add(MaxPool2D())
model.add(Conv2D(24, kernel_size = 3, activation='relu', padding='same'))
model.add(MaxPool2D(padding='same'))
model.add(Conv2D(48, kernel_size = 3, padding='same', activation='relu'))
model.add(MaxPool2D(padding='same'))
model.add(Conv2D(64, kernel_size = 3, padding='same', activation='relu'))
model.add(Flatten())
model.add(Dropout(0.3))
model.add(Dense(256, activation='relu'))
model.add(Dense(1, activation='linear')))

opt =Adam(lr=0.001)
model.compile(optimizer=opt, loss="mean_squared_error", metrics=[r2_keras])

lr_reducer = LearningRateScheduler(lambda x: 0.001 * 0.995 ** x)
EarLY=EarlyStopping(monitor='val_loss', mode='min', min_delta=0, patience
    =50, verbose=0, restore_best_weights=True)

filepath = 'modelFst_.h5'
checkpoint = ModelCheckpoint(filepath, monitor='val_loss', mode='min',
    verbose=1, save_best_only=True)

epochs = 1000

history = model.fit(X_train, Y_train, epochs = epochs,  validation_split =
    0.3,  batch_size=100, callbacks=[lr_reducer, EarLY, checkpoint])
```

Prediction of genetic indexes from image data using this code yielded r^2 values above 0.99 for mAlleles and above 0.98 for F_{ST}.

Fit with data augmentation

In order to get predictions that would perform better with data beyond those used in training we used data augmentation with keras ImageDataGenerator() by rotating and vertically and horizontally shifting the original images. This allows the neural network to learn patterns useful in predicting the response under a larger variety of conditions increasing its ability to generalize.

The code was modified to use this generator for the training set and leaving the validation set without augmentation.

```
datagen = ImageDataGenerator(rotation_range=30, width_shift_range=0.3,
    height_shift_range=0.3,  validation_split=0.3)

Vdatagen = ImageDataGenerator()

bs=100

aigen=datagen.flow(X_train, Y_train, batch_size=bs)
Vaigen=Vdatagen.flow(X_train, Y_train, batch_size=bs)

tsteps =  X_train.shape[0]*0.7/bs
vsteps =  X_train.shape[0]*0.3/bs

history = model.fit(aigen, epochs = epochs,  validation_data = Vaigen,
    verbose=1, steps_per_epoch = tsteps, validation_steps=vsteps, callbacks
    =[lr_reducer, EarLY, checkpoint])
```

This procedure yielded r^2 values ~ 0.8 for both mAlleles and F_{ST}.

Validation of CNN models using simulation software

We obtained synthetic genotype data simulating the microevolutionary dynamics of individuals in a landscape with demographic and movement characteristics trying to resemble as much as possible those known for *Ctenomys australis*. For this we use the program CDPOP (Landguth and Cushman, 2010) which models genetic exchange among individuals as probabilistic functions of movement cost among individuals' locations. These probabilities can be approximated from a layer of resistance to movement between the spatial coordinates of individuals based on characteristics of the habitat where the species lives and the processes of birth, death, dispersal and mating of individuals can be simulated in complex landscapes through time.

To translate the landscape characteristics contained in the information of the satellite image we used the habitat suitability layer obtained with maxent and transformed it into a layer of movement cost using the the mean suitability value of the 16 neighbours of focal pixels (R package gDistance; van Etten, 2017). Then a matrix of least cost path distances between the coordinates of individuals was obtained using the function shortestPath() and lineLength() in package SDraw (McDonald and McDonald, 2020). This yields a distance matrix similar to a matrix of geographic distances, except that for nearby points separated by patches of unsuitable habitat, pairwise distances are increased because the paths between these locations circumvent unsuitable patches. An example of the constrains imposed by the landscape to the movement of individuals is shown in Figure S2 where least cost path lines are demarcated around patches of grasslands, tree plantations or rocky outcrops that are considered as unsuitable environments for tuco-tucos.

R code to obtain the cost distance matrix

```
library(gdistance)
library(SDraw)

Suitable=raster("Suitable.tif")

ft=function(x)mean(x)
cost=transition(x=Suitable, transitionFunction = ft, directions=16)

costDis=matrix(0, length(xytucos),length(xytucos))

for(j in 1:length(xytucos)){
  shp=shortestPath(x=cost, origin=coordinates(xytucos)[j,],
                   goal=coordinates(xytucos)[-j,], output="SpatialLines")
  costDis[j,-j]=lineLength(shp, byid=T)
}
```

Figure S1: Two different landscapes were used during the simulation with CPOP. The first included only the Sample Area (Area1) and the actual coordinates of sampled individuals. The second (Area2), corresponds to a nearby area for which we did not have genetic samples (8 km to the west of our study site. The characteristics of the landscape in this area were used in CDPOP to obtain genotypes and predict F_{st} values with a CNN trained in Area 1.

This cost-distance matrix is used by CDPOP to calculate the probability of gene flow between locations using movement functions to describe how the probability decreases with distance. We used a negative exponential ($p = A \times 10^{-B \times d}$; with A=1 and B=0.01, which reduces probability [p] to extremely low values for cost distances [d] above 500 m). To model the temporal evolution of gene flow with CDPOP, we chose demographic and genetic parameters to match as closely as possible known features of the population biology of *Ctenomys australis* (see Table S1).

We employed two different validation strategies. In the first one, the CDPOP simulations were based on the above-described maxent approximation to the landscape structure of our study area (Area 1 in Figure S1), and considered a total of 112 individuals and 9 loci. Individuals were placed in the actual geographic coordinates. We sampled the 112 genotypes and then evaluated the accuracy of the CNN model trained on the real genetic data and satellite image to predict the genetic structure of this simulated data.

The second type of validation was based on the landscape structure of our study area, but now considered a total of 1000 individuals and 100 loci. Individuals were randomly placed in the landscape using suitability values as probabilities of occurrence. This yielded a spatial distribution resembling that of tuco-tucos; with suitable patches having many

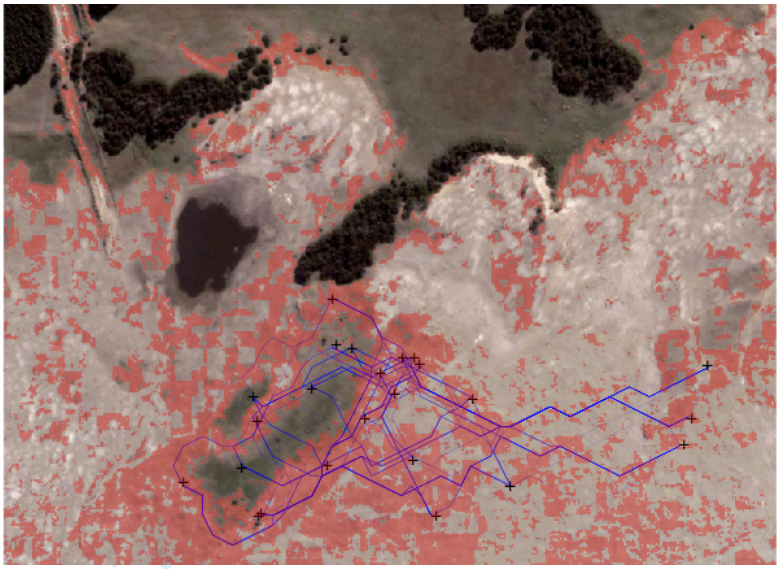

Figure S2: Least cost paths (blue lines) between tuco-tucos' locations (+) used to represent the effect of the landscape on gene flow during simulation of genetic exchange with CDPOP. Light red areas depict the distribution of suitable habitat obtained with maxent.

Table S1: Demographic parameters used during the validation runs with CDPOP.

	Age class			
	0	1	2	3
Male Mortality	20	30	10	100
Female Mortality	20	10	20	100
Mean Fecundity	1	4	4	4
Std Fecundity	1	1	1	1
Male Maturation	0	30	100	100
Female Maturation	0	100	100	100

clustered individuals and with a few scattered individuals located in less suitable areas. We then obtained the cost distance matrix and simulated gene flow as described above. To generate a dataset similar to the real one, we obtained a sample of individuals and loci that matched the numbers in our actual samples. We trained a CNN with the same structure with this synthetic dataset and then transferred what the CNN learned to a nearby area for which we didn't have genetic samples (8 km to the west of our study site, Area 2 in Figure S1). In this new area used the same procedure as before to derive a movement cost layer for CDPOP that corresponded to this new area and run CDPOP simulations with the same parameters values and sampling procedure as described above to obtain a new synthetic dataset and then predicted F_{st} values with the CNN trained with synthetic data in the previous area.

Random Forest and Landscape Metrics to understand fit to genetic indexes

Because the predictive power of deep learning models is somewhat dampened by the impossibility or difficulty in accurately identifying which variables or traits are the drivers of the great predictive capabilities they show, We estimated the distribution of suitable habitat for tuco-tucos in the area using species distribution models using Maxent and sum-

marizing the distribution of patches of suitable habitat with a set of 65 landscape metrics (that measure variation in shape, area, connectivity, etc., see Table S2) and selected a subset that was free of multicollinearity to carry out a Random Forest analysis.

R code to estimate the distribution of suitable area

The rgb image Rast together with geographical coordinates of tuco-tucos tucos obtained during the preprocessing of data were used to obtain a raster layer of suitable habitat for tuco-tucos in the study area.

```
D=distanceFromPoints(Rast, tucos)
# bg points at d from presence points >50 m and  <300m
indi=which(D[]>50 & D[]<300)
D[]=NA
D[indi]=1
bg=data.frame(randomPoints(D, n=1000))
coordinates(bg)=c("x", "y")
proj4string(bg)=proj4string(Rast)

modelHabitat = maxent(x=Rast, p=tucos, a=bg)

# threshold for binary layer
e = evaluate(p=tucos, a=bg, model=modelHabitat, x=Rast)
threshold(e)$kappa

pred=predict(modelHabitat, Rast)

Suitable=pred>threshold(e)$kappa
writeRaster(Suitable, "Suitable.tif", format="GTiff")
```

The geotiff file Suitable.tif is then used to obtain landscape metrics for patches of suitable habitat for the one-hectare squares previously used for training the Convolutional Neural Network.

For example for a one-hectare section delimited by x coordinates from 339680.13 to 339780.13 and y coordinates from -4276704 to -4276604

R code to obtain Landscape Metrics of suitable habitat

```
library(landscapemetrics)

Suitable=raster("Suitable.tif")

MetricsLandscape=function(landscape){
  df[1, 1] =lsm_l_ai(landscape)$value
  df[1, 2] =lsm_l_area_cv(landscape)$value
  df[1, 3] =lsm_l_area_mn(landscape)$value
  df[1, 4] =lsm_l_area_sd(landscape)$value
  df[1, 5] =lsm_l_cai_cv(landscape)$value
  df[1, 6] =lsm_l_cai_mn(landscape)$value
  df[1, 7] =lsm_l_cai_sd(landscape)$value
  df[1, 8] =lsm_l_circle_cv(landscape)$value
  df[1, 9] =lsm_l_circle_mn(landscape)$value
  df[1, 10] =lsm_l_circle_sd(landscape)$value
  df[1, 11] =lsm_l_cohesion(landscape)$value
  df[1, 12] =lsm_l_condent(landscape)$value
  df[1, 13] =lsm_l_contag(landscape)$value
  df[1, 14] =lsm_l_contig_cv(landscape)$value
  df[1, 15] =lsm_l_contig_mn(landscape)$value
```

```
  df[1, 16] =lsm_l_contig_sd(landscape)$value
  df[1, 17] =lsm_l_core_cv(landscape)$value
  df[1, 18] =lsm_l_core_mn(landscape)$value
  df[1, 19] =lsm_l_core_sd(landscape)$value
  df[1, 20] =lsm_l_dcad(landscape)$value
  df[1, 21] =lsm_l_dcore_cv(landscape)$value
  df[1, 22] =lsm_l_dcore_mn(landscape)$value
  df[1, 23] =lsm_l_dcore_sd(landscape)$value
  df[1, 24] =lsm_l_division(landscape)$value
  df[1, 25] =lsm_l_ed(landscape)$value
  df[1, 26] =lsm_l_enn_cv(landscape)$value
  df[1, 27] =lsm_l_enn_mn(landscape)$value
  df[1, 28] =lsm_l_enn_sd(landscape)$value
  df[1, 29] =lsm_l_ent(landscape)$value
  df[1, 30] =lsm_l_frac_cv(landscape)$value
  df[1, 31] =lsm_l_frac_mn(landscape)$value
  df[1, 32] =lsm_l_frac_sd(landscape)$value
  df[1, 33] =lsm_l_gyrate_cv(landscape)$value
  df[1, 34] =lsm_l_gyrate_mn(landscape)$value
  df[1, 35] =lsm_l_gyrate_sd(landscape)$value
  df[1, 36] =lsm_l_iji(landscape)$value
  df[1, 37] =lsm_l_joinent(landscape)$value
  df[1, 38] =lsm_l_lpi(landscape)$value
  df[1, 39] =lsm_l_lsi(landscape)$value
  df[1, 40] =lsm_l_mesh(landscape)$value
  df[1, 41] =lsm_l_msidi(landscape)$value
  df[1, 42] =lsm_l_msiei(landscape)$value
  df[1, 43] =lsm_l_mutinf(landscape)$value
  df[1, 44] =lsm_l_ndca(landscape)$value
  df[1, 45] =lsm_l_np(landscape)$value
  df[1, 46] =lsm_l_pafrac(landscape)$value
  df[1, 47] =lsm_l_para_cv(landscape)$value
  df[1, 48] =lsm_l_para_mn(landscape)$value
  df[1, 49] =lsm_l_para_sd(landscape)$value
  df[1, 50] =lsm_l_pd(landscape)$value
  df[1, 51] =lsm_l_pladj(landscape)$value
  df[1, 52] =lsm_l_pr(landscape)$value
  df[1, 53] =lsm_l_prd(landscape)$value
  df[1, 54] =lsm_l_rpr(landscape)$value
  df[1, 55] =lsm_l_shape_cv(landscape)$value
  df[1, 56] =lsm_l_shape_mn(landscape)$value
  df[1, 57] =lsm_l_shape_sd(landscape)$value
  df[1, 58] =lsm_l_shdi(landscape)$value
  df[1, 59] =lsm_l_shei(landscape)$value
  df[1, 60] =lsm_l_sidi(landscape)$value
  df[1, 61] =lsm_l_siei(landscape)$value
  df[1, 62] =lsm_l_split(landscape)$value
  df[1, 63] =lsm_l_ta(landscape)$value
  df[1, 64] =lsm_l_tca(landscape)$value
  df[1, 65] =lsm_l_te(landscape)$value
  return(df)
}

e=extent(c(339680.13, 339780.13, -4276704, -4276604))

landscape=crop(Suitable, e)
proj4string(landscape)=CRS("+proj=utm +zone=21 +datum=WGS84")

extent(landscape)=e

MetricsLandscape(landscape)
```

MetricsLandscape(landscape) gives 65 landscape metrics for the example one-hectare section of landscape.

These metrics are then calculated for all one-hectare sections and saved to a csv file. The file LSMetrics.csv contains genetic indexes and 65 landscape metrics for all one-hectare squares. To use these variables in a Random Forest model a subset of 19 variables with pearson correlation below 0.8 was identified with the following R script.

```
library(caret)
x=LSMetrics[,-(1:2)]
# leave out variables without variation
indi=which(apply(x, 2, sd)!=0)
x=x[,indi]
corr_matrix=cor(x)
highCorr=findCorrelation(x=corr_matrix, cutoff=0.8)
cm=cor(x[,-highCorr])
#get the names of uncorrelated variables
colnames(cm)
```

With the subset of uncorrelated variables we trained a Random Forest model to predict the genetic indexes and evaluate how different landscape metrics contribute to the predictions of the model using shapley values. We illustrate the procedure for mAlleles.

Python code Random Forest model

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
from sklearn.ensemble import RandomForestRegressor
import seaborn as sbn

df = pd.read_csv('LSMetrics.csv')

# Landscape metrics with pairwise corrleation below 0.8

vars = ["lsm_l_area_mn", "lsm_l_area_sd", "lsm_l_cai_cv", "lsm_l_circle_mn"
    , "lsm_l_circle_sd",  "lsm_l_cohesion", "lsm_l_contig_cv",
         "lsm_l_contig_sd", "lsm_l_dcad" , "lsm_l_dcore_cv",  "lsm_l_enn_cv
    ", "lsm_l_enn_mn", "lsm_l_frac_sd", "lsm_l_gyrate_cv",
         "lsm_l_lpi" , "lsm_l_mutinf",        "lsm_l_pafrac", "
    lsm_l_shape_cv", "lsm_l_shape_mn"]

X =  df[vars]

# The target variable is 'MAlleles'
Y = df['MAlleles']

np.random.seed(0)
# Split the data into train and test data:
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.3)

model = RandomForestRegressor(max_depth=100, random_state=0, n_estimators
    =10000)
model.fit(X_train, Y_train)
```

This provides a r^2=0.968 for the validation dataset. To get a hint of how the landscape variables contribute to the prediction of mAlleles we use shapley values.

Python code for Shapley values

```
import shap
shap_values = shap.TreeExplainer(model).shap_values(X)
vals= np.abs(shap_values).mean(0)
```

Shapley values are now available in shap_values and the contribution of landscape variables are in vals. Shapley plots to inspect the contributions can be obtained using shap.dependence_plot() ; for example:

```
shap.dependence_plot("lsm_l_shape_cv", shap_values, X)
```

Transfer of predictions to the whole study area

The weights of the CNN model saved in modelFst_.h5 (see above) can be loaded and used to predict the spatial distribution of F_{ST} values in the study area. This is straightforward with model.predict() in keras. Loading the jpg image of the study area (or other areas for that matter) we can get predictions for different one-hectare cuts.

```
model.load_weights('modelFst_.h5')
from matplotlib import image
from matplotlib import pyplot

# load image as pixel array
image = image.imread('area.jpg')

# random xy cells of the image
xl=np.random.randint(117,image.shape[0]-117)
xu=xl+117
yl=np.random.randint(117,image.shape[1]-117)
yu=yl+117

# get a random 117 by 117 crop of the image
imgc=image[xl:xu,yl:yu,:]
# display the array of pixels as an image
pyplot.imshow(imgc)
pyplot.show()

x_array=imgc/255

print(model.predict(x_array.reshape((1,117,117,3))))
```

Table S2: Landscape metrics used to characterize the distribution of suitable habitat in *Ctenomys australis*. A full description for the metrics and their equations is available at https://www.rdocumentation.org/packages/landscapemetrics.

Metric name	Type of metric
area_cv	area and edge metric
area_mn	area and edge metric
area_sd	area and edge metric
ed	area and edge metric
gyrate_cv	area and edge metric
gyrate_mn	area and edge metric
gyrate_sd	area and edge metric
lpi	area and edge metric
ta	area and edge metric
te	area and edge metric
cai_cv	core area metric
cai_mn	core area metric
cai_sd	core area metric
core_cv	core area metric
core_mn	core area metric
core_sd	core area metric
dcad	core area metric
dcore_cv	core area metric
dcore_mn	core area metric
dcore_sd	core area metric
ndca	core area metric
tca	core area metric
circle_cv	shape metric
circle_mn	shape metric
circle_sd	shape metric
contig_cv	shape metric
contig_mn	shape metric
contig_sd	shape metric
frac_cv	shape metric
frac_mn	shape metric
frac_sd	shape metric
pafrac	shape metric
para_cv	shape metric
para_mn	shape metric
para_sd	shape metric
shape_cv	shape metric
shape_mn	shape metric
shape_sd	shape metric
ai	aggregation metric
cohesion	aggregation metric
contag	aggregation metric
division	aggregation metric
enn_cv	aggregation metric
enn_mn	aggregation metric
enn_sd	aggregation metric
iji	aggregation metric
lsi	aggregation metric
mesh	aggregation metric
np	aggregation metric

Table S2: Continued.

Metric name	Type of metric
pd	aggregation metric
pladj	aggregation metric
split	aggregation metric
condent	complexity metric
ent	complexity metric
joinent	complexity metric
mutinf	complexity metric
msidi	diversity metric
msiei	diversity metric
pr	diversity metric
prd	diversity metric
rpr	diversity metric
shdi	diversity metric
shei	diversity metric
sidi	diversity metric
siei	diversity metric